

Automated construction and graphical presentation
of protein blocks from unaligned sequences

Steven Henikoff¹, steveh@howard.fhcrc.org, Ph: (206) 667-4515, FAX: (206) 667-5889

Jorja G. Henikoff, jorja@howard.fhcrc.org, (206) 667-4515

William J. Alford², billa@cs.wisc.edu, (608) 262-6600

Shmuel Pietrokovski, pietro@sparky.fhcrc.org, (206) 667-4509

Basic Sciences Division

¹Howard Hughes Medical Institute

Fred Hutchinson Cancer Research Center

Seattle, Washington 98104, USA

²Present address: Department of Computer Science, University of Wisconsin,

Madison, WI 53706, USA

Keywords: protein blocks, multiple sequence alignment, database searching, Gibbs sampling,
sequence logos, World Wide Web, Tc1 transposon family

SUMMARY

Protein blocks consist of multiply aligned sequence segments that correspond to the most highly conserved regions of protein families. Typically, a set of related proteins has more than one region in common and their relationship can be represented as a series of ungapped blocks separated by unaligned regions. Blockmaker is an automated system available by electronic mail (blockmaker@howard.fhcrc.org) and the World Wide Web (<http://www.blocks.fhcrc.org>) that finds blocks in a group of related protein sequences submitted by the user. It adapts and extends existing algorithms to make them useful to biologists looking for conserved regions in a group of related proteins sequences. Two sets of blocks are returned, one in which candidate blocks are detected using the MOTIF algorithm and the other using a Gibbs sampler algorithm that has been adapted for full automation. This use of two block-finding methods based on completely different principles provides a "reality check", whereby a block detected by both methods is considered to be correct. Resulting blocks can be displayed using the information-based "sequence logo" method, adapted to incorporate sequence weights, which provides an intuitive visual description of both the residue and the conservation information at each position. Blocks generated by this system are useful in diverse applications, such as searching databases and designing degenerate PCR primers. As an example, blocks made from amino acid sequences related to *Caenorhabditis elegans* Tc1 transposase were used to search GenBank, revealing that several fish and amphibian genomic sequences harbor previously unreported Tc1 homologs.

INTRODUCTION

As a result of the exponential growth of sequence databanks, it is thought that most protein families are represented by one or more known sequences [1]. It is further estimated that nearly all ancient conserved regions, defined as aligned sets of related proteins that cross phylum boundaries, have been represented in sequence databanks for years [2, 3]. This accelerating availability of multiple members of so many protein families focuses attention on the need for methods that find alignments between multiple family members and that allow extraction of interesting and useful information from such alignments. Traditional multiple alignment methods that are useful in evolutionary studies [4-9] are not always ideal for applications that focus on conserved regions of proteins, such as searching databases [10, 11], designing degenerate PCR primers [12] and designing peptides for raising antibodies [13]. For these applications, short ungapped segments representing the most highly conserved regions of proteins, referred to as "blocks" [14, 15], provide a useful approach to multiple alignment and information problems, especially where the only detectable relationships between distant relatives consist of short motifs. An increasing number of blocks-based methods are now available [11, 16-22], including searchable databases that themselves consist of blocks [16, 18] or other motif representations [23-26].

A common problem that faces a biologist interested in a new protein family is that motif representations are not available for it, or that the ones that are available lack very recently described members, or the members are too broadly classified. This situation requires that the biologist carry out a multiple alignment procedure with selected members, and several multiple alignment tools are freely available for this purpose, such as CLUSTAL [5, 9] and MACAW [27]. However, these programs have limitations when applied to motif identification. On the one hand, fully automated programs such as CLUSTAL are designed to align sequences from one end to another, not to identify motifs within sequences that might share only one or a few conserved regions. On the other hand, MACAW is well suited for identification of motifs, but it is interactive, not automated, and it has time requirements that become prohibitive for more than a dozen or so sequences. Full automation and reasonable time requirements are important if a motif-finding program is to run over the Internet on a central server.

A program that meets the above requirements is the two-step PROTOMAT system [16], in which a motif finder [15] is combined with a block assembler to provide a set of blocks corresponding to the conserved regions of proteins. Here we describe an extension of PROTOMAT, Blockmaker, accessible either by electronic mail or over the World Wide Web (WWW), which returns blocks constructed from any submitted set of protein sequences. To assist the user in determining the significance of the blocks it returns, Blockmaker constructs blocks using two different motif-finding algorithms. One is a motif finder based on spaced triplets [15], and the other is a fully automated adaptation of a motif finder based on Gibbs sampling [20]. Both algorithms assume that all sequences in a group share at least one motif, and so the programs continue until they find at least one, even in randomly generated sequences. This feature is undesirable in the context of an Internet server, since a user can be misled into concluding that a sequence contains a motif when the reported alignment is spurious. Unfortunately, statistical solutions that are adequate for pairwise alignment methods

[28] are not directly generalizable to multiply aligned sequences, and conservative rules of thumb must be applied [27]. If both algorithms find similar blocks, however, the user can be more confident about them.

The WWW implementation of Blockmaker allows the user to examine returned blocks utilizing a modification of the intuitive logo display method of Schneider and Stephens [29]. This modification applies sequence weighting to compensate for undesirable redundancy [30-32] which is a common characteristic of sequence families.

As an example of the practical use of the Blockmaker system, we show its application to the family of Tc1 transposase proteins [33], where the resulting blocks are used to search GenBank, revealing previously unreported examples of related family members.

METHODS

(a) The PROTOMAT block-finding system

As originally described [16], the first step of the two-step PROTOMAT system consists of a modified version of the MOTIF program of Smith *et al.* [15]. MOTIF looks for the presence of all spaced triplets out to a maximum distance in at least a subset of sequences. For example, one spaced triplet is Ala-Ala-Ala, another is Ala-x-Ala-Ala and another is Val-x-x-x-Ala-x-x-x-x-x-x-Cys where x represents any amino acid. A spaced triplet found in enough sequences provides an alignment against which the sequences lacking the triplet are aligned to maximize a block score, which is determined using an amino acid substitution matrix (currently BLOSUM 62 [37]). MOTIF is applied using parameters that are selected automatically to yield large numbers of candidate blocks. The MOTIF "repeats" parameter is set to zero.

In the second step, a graph theory method (MOTOMAT) is used to determine a best set of blocks, which is the highest scoring set of blocks with at least a minimum number of sequences, where for all sequences retained, the blocks must be in the same order without overlapping. MOTOMAT 1) merges overlapping candidate blocks, 2) extends alignments to provide blocks with maximum scores, and 3) determines the best set of blocks, where the blocks are all in the same order and do not overlap for a significant number (determined empirically by MOTIF) of the sequences in the group. MOTOMAT does not realign sequences that fail to conform, but rather discards them. We have found this procedure to be very effective in finding blocks for even the most distantly related families, and this automated system is the basis for the current Blocks Database [39]. This version of PROTOMAT is referred to as "MOTIF", and provides one set of blocks reported by Blockmaker.

To provide a second set of blocks, we have developed a new version of PROTOMAT, referred to as "GIBBS". An extension of the iterative Gibbs sampling program of Lawrence *et al.* [20] is substituted for the MOTIF program, such that candidate blocks are similarly delivered to MOTOMAT. In each step, random starting positions are chosen to give a block for all but one of the sequences. This remaining sequence is slid along the block to score each segment based on information content. The probability that any starting position in this sequence will be chosen for the next step is proportional to the segment score. This procedure is reiterated a large number of times until the score is maximized. False starts will

fail to improve, whereas detection of a true pattern in even a small subset of sequences leads to rapid improvement in score.

The original Gibbs sampling program searches for a user-specified number of blocks (N) of user-specified width (W). Our extension consists of a heuristic that uses information obtained from the sequence set to arrive at N, then runs the Gibbs program with an assortment of widths. For every run, the number of blocks in each sequence is specified, and only one site per sequence is allowed for each block [20]. The ordering option is not used. The Gibbs program is run multiple times until all possible widths between the default minimum and maximum have been tried, resulting in a set of candidate blocks, one for each allowable width. Since multiple runs of the Gibbs program become computationally demanding relative to MOTIF runs, we limit the number of blocks sought and the number of runs. As implemented, the heuristic chooses N based on the width of the shortest sequence. Table 1 shows the current defaults for sequence sets with different minimum lengths. This procedure is based in part on the rationale that all blocks will have to fit within the shortest sequence in the set, and that a short sequence is unlikely to contain as many blocks as a long sequence, and in part on empirical testing on a variety of families using different combinations of N and W (data not shown). For GIBBS, the MOTOMAT significance level (minseq) is determined from a regression equation fit to data from version 7.01 of the Blocks Database as follows: for fewer than 9 sequences, $\text{minseq} = \text{number of sequences}$; otherwise, $\text{minseq} = 0.47 * (\text{number of sequences}) + 3.55$.

(b) Sequence logos

The MAKELOGO program [29] provides a graphical display of a multiple alignment consisting of ordered stacks of letters representing amino acids at successive positions. The height of a letter in a stack increases with increasing frequency of the amino acid, and the height of a stack of letters increases with increasing conservation of the aligned position. Letters in stacks with single residues are taller than those in stacks with multiple residues (Fig. 2). This is because the height of an amino acid at a position increases as its representation increases in the block and as the position becomes increasingly conserved [29]. Within stacks, the most highly represented residues are not only taller, they also lie higher in the stack, so that the most prominent residue at the top is also the one predicted to be the most likely to occur at that position. Residues at each position are color-coded based on amino acid properties and can be perceived at a glance.

We modified the program to accept a position-specific scoring matrix (PSSM), often referred to as a profile [34], computed from each block. For each column of a PSSM, corresponding to a position in the block alignment, a set of sequence-weighted counts representing the frequency of occurrence of each amino acid is divided by the corresponding set of expected frequencies obtained from a protein sequence database [10], resulting in odds ratios. We use position-based weights [32] for sequence weighting and SWISS-PROT v. 26 [35] amino acid frequencies for expected frequencies. This procedure reduces the tendency for sequence sub-families over-represented in the block to dominate stacks and increases the representation of rare amino acids in a stack relative to common amino acids. In a logo, a "strong" highly conserved block can be readily distinguished from a "weak" (perhaps questionable) block by the heights and densities of the stacks. The most conserved regions

within blocks can also be readily perceived.

(c) Searching sequence databases with blocks

Blocks produced by Blockmaker can be used to search a sequence database to find additional family members [10, 11, 33]. Although this service is not provided by Blockmaker because of practical limitations, the BLIMPS searching program and the MULTIMAT search results analyzer (written in the C programming language) are available by anonymous ftp (from howard.fhcr.org). Each block for a family is searched independently against a sequence database by BLIMPS. BLIMPS converts the query block to a PSSM, which is used to score all segments the width of the block [36]. Each segment in the database is scored by aligning the PSSM with the segment and adding the scores obtained for each position in the segment. Scoring starts at the beginning of each sequence, the PSSM is then slid over one residue and the process is repeated, and so on until all segments in the database have been scored and the top-scoring alignments saved. For a DNA database, each sequence is searched in all six frames (or optionally in just three frames on one strand).

The program MULTIMAT combines the BLIMPS search results for all blocks from a family against a sequence database and compares the highest-scoring alignments with the query blocks. If more than one block from the family detects the same database sequence and the distance between the blocks is consistent with the distance between them in the sequences used to make the blocks, then a P-value is computed for the multiple block hit as described [33]. For a DNA database, multiple block hits can occur in different frames on the same strand.

RESULTS

(a) Block-finding algorithms used by Blockmaker

Blockmaker uses the two-step PROTOMAT system for finding a best set of blocks representing a group of related proteins. The first step finds candidate alignments and the second step extends the alignments and sorts them in such a way that a best set is chosen. Since 1991, the first step has employed a modified version of the MOTIF program of Smith et al. [15] which looks for spaced triplets of amino acids. To maximize the sensitivity of MOTIF, parameters are chosen so that some spaced triplets are found even for shuffled sequences. However, MOTIF may still fail to detect blocks that lack a spaced triplet within the minimum number of sequences [*e. g.* 38].

A completely different approach to the block-finding problem uses an iterative "Gibbs sampling" strategy [20]. There are serious limitations to the original implementation of this strategy, in that the number of blocks representing a family and the width of each of the blocks must be specified in advance for each run. For typical families, which might have several blocks of different widths, it is extremely impractical to try all possible combinations of number of blocks and widths. In addition, the Gibbs sampler was demonstrated only on families that were first purged of all but the most distant relatives [20], so that its performance on typical groups, which include a mixture of close and distant relatives, was unknown.

In spite of its limitations, the Gibbs sampling strategy is attractive because it

complements MOTIF. GIBBS can find motifs that do not have a spaced triplet, and it uses information content instead of a substitution matrix to score alignments. Therefore, we have investigated its use as a motif finder to provide candidate blocks for MOTOMAT to extend and assemble. We have developed an effective heuristic strategy for doing this, which requires only a small number of runs of the Gibbs sampling program and which can be carried out in a reasonable amount of time (Table 1). This strategy inevitably makes compromises not necessary with MOTIF (which is more exhaustive with respect to block width and number) and sometimes misses blocks that MOTIF finds (although the opposite can occur). Furthermore, GIBBS is much slower than MOTIF.

We have constructed Blocks Databases by successive application of the both the MOTIF and GIBBS versions of PROTOMAT to 698 unique groups catalogued in Prosite 11.0. Group size ranged from 2 to 384 sequences after removal of fragment sequences and those with identical Swiss-Prot protein names and first 3 characters of the organism name. This pruning retains redundant full-length sequences that are paralogs within a species and are orthologs above the species level. Overall, the resulting GIBBS blocks were comparable to those generated using MOTIF for the same Prosite groups. MOTIF found 2679 blocks for the 698 groups, while GIBBS found 2177, as could be anticipated from the restrictions of the GIBBS heuristic. On average, both the GIBBS blocks and the MOTIF blocks included about 96% of the sequences in a group, and both scored an average of 98% of the sequences in the group above 99.5% of other sequences in searches against Swiss-Prot 27.

In some cases, more optimally aligned blocks were obtained with GIBBS. For example, although GIBBS and MOTIF found the same five blocks characteristic of the cytosine methylase family (14), MOTIF found them in only 27 of the 31 sequences, whereas GIBBS found them in all but one sequence. The missed sequence, mouse cytosine methyltransferase, was later shown to be frameshifted such that it lacked one of the blocks, and as a result the sequence was discarded by MOTOMAT. These results confirm that GIBBS works well in practice for even very large families. In addition, they demonstrate that GIBBS provides good blocks for protein families that have not been purged of redundant sequences. Both the MOTIF and GIBBS databases are available upon request.

The complementary strengths and weaknesses of MOTIF and GIBBS block-finding algorithms suggest that they can be compared to provide a "reality check". Blockmaker will *always* report blocks, even if random sequences are provided. Because of this, the ability of Blockmaker to find blocks should not ordinarily be interpreted as evidence for homology, although the blocks can aid in the detection of conserved regions and in the determination of family relationships. We find that if sequences truly have conserved regions in common, then both runs yield similar, and sometimes identical sets of blocks. However, if sequences have nothing in common, we find that the two block-finding algorithms pick up completely different meaningless blocks.

We have not automated the comparison of the two sets of blocks. To do so requires subjective criteria with respect to what degree of difference is tolerable. In practice, deciding which blocks are similar based on the text display is usually a simple task, for example in Fig. 1 (below). Moreover, the sequence logo option can aid in judging the reality of any block not found using both methods.

(b) Sending sequences to Blockmaker

The Blockmaker servers accept a minimum of 2 and a maximum of 250 protein sequences received either by e-mail (blockmaker@howard.fhcrc.org) or through the WWW (<http://www.blocks.fhcrc.org/>). Each server will run the PROTOMAT system using MOTIF and GIBBS. Blockmaker is especially effective for large numbers of sequences that are difficult to align by standard multi-sequence alignment methods, but is not a very good method for aligning just two sequences or multiple sequences that are all very similar to one another. A fragmentary sequence lacking a probable conserved region should not be submitted to Blockmaker, because this could cause the system to reject otherwise correct blocks that lack the sequence.

Sequences must be in a single format, one after the other in the body of the message. FASTA and other common formats are accepted.

(c) Results returned by Blockmaker

A typical Blockmaker run requires a few minutes, with the time increasing approximately in proportion to the total number of amino acids present in all of the protein sequences submitted. Fig. 1 shows the results of a Blockmaker run, where the input consisted of 9 known full-length members of the Tc1 family of transposons. In this case, MOTIF reported 4 blocks and GIBBS reported 5 blocks including all 9 sequences. All 4 MOTIF blocks correspond closely to the first 4 GIBBS blocks, with minor differences only in how far the blocks extend to one side or the other for Blocks A and D. We conclude that these blocks are correct, noting that some misalignments near the edges might be tolerated for either MOTIF or GIBBS, and this could account for the minor differences. Examination of the last GIBBS block reveals that it is very likely correct: although the segments appear to be quite dissimilar, and just one position contains an invariant residue, the distances to the previous block are within a range of only 17-23 amino acids for all 9 sequences. It appears that MOTIF misaligned the two most diverged sequences (EsTet1 and DmHB1) in this region, and as result, discarded this block from the best set because the order of blocks along the sequence was inconsistent with the order for the other 7 sequences (data not shown). However, GIBBS found what appears to be the correct alignment for both of these segments with the others, and as a result, the order of blocks for these two diverged sequences was consistent with the order for the others, so the block was retained in the best set. Note that even though GIBBS performed better than MOTIF in this case because it optimized a difficult alignment, in other cases GIBBS can miss blocks, possibly because of limitations in the heuristic. Therefore, we recommend that all major differences between GIBBS and MOTIF blocks be treated with caution, using other clues to guide one's judgment, such as biochemical data and distances between blocks. For another example of this strategy, see [38].

It is important to realize that whereas blocks can be extremely useful for multiple sequence alignments (*e.g.*, the manually-assisted MACAW program [27] is based on finding blocks), PROTOMAT was not specifically designed for this purpose, but rather to find blocks that are effective for searching applications when provided with groups of proteins that are known to have motifs in common. In this context, it is more important to find real motifs and avoid finding spurious ones than to accurately detect these motifs in all of the sequences

that comprise the group. As a result, PROTOMAT-generated blocks occasionally exclude sequences that do not conform with the others or include these sequences, but with misalignments, especially near the edges of blocks. These errors can be tolerated so long as the block is correctly identified for the large majority of sequences; in such a case the contributions from misaligned segments will be diluted out, and so searching performance will be affected only slightly. The Blockmaker WWW page provides links to other sites for users wishing to obtain a gapped global multiple alignment.

(d) Logos from blocks

Since blocks are multiple alignments, they may be too large or complicated for intuitive evaluation. This situation has encouraged simplification of the information using text-based representations variously referred to as patterns, signatures or consensus sequences [23]. Unfortunately, patterns oversimplify the rich information in a block. A more useful representation that conveys the information in a block without discarding anything is sequence logos [29], which consist of color-coded stacks of letters that show the contribution of each amino acid at each position. The WWW version of Blockmaker provides a "logos" option.

A particular advantage of viewing logos as opposed to viewing blocks directly is that blocks with many sequences can be viewed in a compact form. This is particularly important for the Blockmaker system, which can find blocks in families consisting of hundreds of sequences. However, as described [29], logos are based on residue frequencies, even though these frequencies might be strongly biased because of sequence redundancy in the block. This problem can be especially severe where a family consists of very similar mammalian sequences and a few diverged microbial sequences. We have addressed this problem by modifying the sequence logo method to display odds ratios from a position-specific scoring matrix, incorporating sequence weights to compensate for redundancy [32]. For example, the first column in GIBBS Tc1 Block D (Fig. 1) includes 6 leucines, mostly from closely-related *C. elegans* members, and 2 methionines from more distant relatives; nevertheless, the sequence-weighted logo (Fig. 2) places methionine at the top, primarily because the 2 methionines are from highly informative sequences, whereas the 6 leucines are from relatively redundant sequences.

(e) Searching sequence databases with blocks

One application of blocks is to use them to find distant relatives in sequence databases. Although this procedure becomes computationally intensive for a database the size of GenBank, which requires twice as many comparisons as there are residues for 6-frame translation, it often yields new family members, especially when the database sequence is incomplete or erroneous. Families enlarged in this way can be sent to Blockmaker to get refined blocks and the search is then repeated [10, 11]. In an earlier study of the Tc1 transposase family [33], multiple blocks representing the 6 full-length sequences known at the time were used to search GenBank 70, leading to the discovery of the first vertebrate (a fish) homolog of a DNA-based transposon, as well as the discovery of the first relationship between a eukaryotic and a prokaryotic transposon. An update of this search has been carried out three years later using the present set of (GIBBS) Tc1 blocks representing 9 sequences

(Fig. 1) and GenBank 87, with the results shown in Table 2. Multiple-block hits were assessed by asking whether any of the 100 top-ranking blocks was supported by the detection of one or more lower-ranking blocks among the top 1000.

The top-ranking 58 block hits were judged to be true positives, mostly known Tc1 family members. A total of 9 multiple-block hits were judged to be new discoveries; all of these ranked above 135, whereas the first false positive multiple-block hit ranked 552. Interestingly, 5 of the 9 newly discovered Tc1 elements are present in fish genomic sequences, and all previous vertebrate Tc1 sequences were detected in fish sequences [33, 40, 41]. Considering that no Tc1 elements have yet been detected in mammals, fish appear to be extraordinarily rich in Tc1 elements. In addition, two *Xenopus* genomic sequences were found to harbor Tc1 elements, evidently the first such examples in an amphibian. Whether the presence of Tc1-like transposons in the oldest vertebrate lineages but not evidently in more recent lineages results from host differences or ancestral loss is not clear. But until more is known, it would be prudent to assume that higher vertebrates are potentially susceptible to Tc1 transposons and might lack host defenses. The rapid spread of the P-transposon into all known natural populations of *Drosophila melanogaster* during recent decades [42] demonstrates the virulence of a very similar genetic parasite.

For one of the *Xenopus* hits, the MULTIMAT output is shown (Fig. 3). MULTIMAT puts together blocks that align with translated segments even when they are out of frame with one another. Here, Block B is in a different frame from Blocks A and E, yet the interblock distances are very similar to what has been seen in Tc1 transposase, revealing that frameshifts on either side of the segment aligning with Block B have occurred. These frameshifts suggest that the hit is to a pseudogene, which is typical of Tc1 family members, although sequencing errors are not ruled out. For each of the three *Xenopus* segments detected, the most similar segment found in the block differed from block to block (DmBari-1 from Block A, TC1A_CAEBR from Block B and DhMinos from Block C). This illustrates how information from multiply aligned sequences can aid in the detection of distant relationships that would be more difficult to detect using any single sequence. It is perhaps significant that none of the three block segments closest to the *Xenopus* segment was from the input vertebrate sequence (EsTes1 from hagfish), consistent with the notion that Tc1-mariner transposons had diversified long before the appearance of vertebrates [33, 41, 43, 44].

An interesting single block hit above all false positives is a repeated sequence from *Rhizobium* bacteria (GenBank accession # RFU18764), suspected to be a transposable element, but not previously known to be related to Tc1. In all, 10 very likely new Tc1 family members were identified in this search.

At the time of the earlier study [33], homology between Tc1 transposons and mariner transposons had not been recognized. Neither of the two mariner sequences present in GenBank 70 were detected at that time. However, homology between Tc1 and mariner is now well established [37, 43, 45, 46] and large numbers of mariner sequences, mostly from insects, have been amplified and sequenced [44, 45]. As a result of the presence of so many mariner sequences, several were detected in the present search, including multiple-block hits involving GIBBS Blocks C, D and E (Table 2). For single block hits, there were 10 mariner sequences above the first likely false positive (rank 59). Most of these are ~220 bp coding sequences that were amplified from different transposons present in the hornfly, *Haematobia*

irritans. For multiple-block hits, there were 4 mariner hits (one a new discovery) above the first likely false positive with multiple blocks (rank 552).

CONCLUSIONS

The Blockmaker WWW and electronic mail servers adapt and extend existing algorithms to identify and display conserved regions shared by a family of proteins of interest for the biologist. We have introduced a fully automated implementation of a block finder based on Gibbs sampling. Together with our previous automated system, Blockmaker provides a "reality check" strategy, in which two different block finders using different scoring schemes provide independent sets of results for comparison. Blocks found by both methods are considered confirmed, whereas blocks found by only one method require further evidence for confirmation. For example, the Tc1 E block found only by GIBBS was confirmed because the distance from the previous block was found to be very similar for all sequences. Lack of confirmation might indicate that the block is not to be trusted. This reality check feature has proven useful in a recently published study [47]. Blockmaker was used to analyze the suspected relationship between the yeast chromosome segregation protein, Mif2, and blocks from animal centromeric and other DNA-associated proteins. Only two of the 4-5 blocks obtained using MOTIF and GIBBS were in common, in support of inferences based on pairwise methods that these two regions are truly homologous, whereas the other blocks reflect marginal similarity that might not be meaningful.

We expect that the utility of blocks for analysis of conserved regions of proteins, such as for designing degenerate PCR primers [12], will increase as protein families grow in size and diversity. The ability of Blockmaker to align hundreds of sequences automatically and to display blocks as sequence logos based on sequence-weighted odds ratios should prove especially useful for analyzing domains and conserved regions from large families. In contrast, some other block-finding methods, such as MACAW [27], limit the number of sequences and require user input to arrive at a final set of blocks. Blockmaker should also prove useful with diverged families that have only short conserved regions, a situation that is especially challenging for full multiple alignment methods, such as CLUSTAL [9]. Together with the companion Blocks server [39] for searching, retrieval and viewing of the Blocks Database, Blockmaker can provide useful information on conserved regions of proteins for drawing biological inferences.

ACKNOWLEDGEMENTS

We thank Megan Brown for sharing results prior to publication and for commenting on the system during its development. S. H. also thanks Alan Christiansen for drawing attention to the possible involvement of genomic parasites in extinction events. This work was supported by a grant from NIH (GM29009).

REFERENCES

- [1] Chothia C. (1992) Nature 357, 543.
- [2] Green P., Lipman D., Hillier L., Waterston R., States D. and Claverie J.-M. (1993)

- Science 259, 1711.
- [3] Green P. (1994) *Curr. Opin. Struct. Biol.* 4, 404.
 - [4] Feng D.F. and Doolittle R.F. (1987) *J. Mol. Evol.* 25, 351.
 - [5] Higgins D.G. and Sharp P.M. (1988) *Gene* 73, 237.
 - [6] Corpet F. (1988) *Nucleic Acids Res.* 16, 10881.
 - [7] Lipman D.J., Altschul S.F. and Kececioglu J.D. (1989) *Proc. Natl. Acad. Sci. USA* 86, 4412.
 - [8] Hein J. (1990) *Meth. Enzymol.* 183, 626.
 - [9] Thompson J.D., Higgins D.G. and Gibson T.J. (1994) *Nucleic Acids Res.* 22, 4673.
 - [10] Henikoff S., Wallace J.C. and Brown J.P. (1990) *Meth. Enzymol.* 183, 111.
 - [11] Tatusov R.L., Altschul S.F. and Koonin E.V. (1994) *Proc. Natl. Acad. Sci. USA* 91, 12091.
 - [12] D'Esposito M., Pilia G. and Schlessinger D. (1994) *Hum. Mol. Genet.* 3, 735.
 - [13] Close T.J., Fenton R.D. and Moonan F. (1993) *Plant Mol. Biol.* 23, 279.
 - [14] Posfai J., Bhagwat A.S., Posfai G. and Roberts R.J. (1989) *Nucleic Acids Res.* 17, 2421.
 - [15] Smith H.O., Annau T.M. and Chandrasegaran S. (1990) *Proc. Natl. Acad. Sci. USA* 87, 826.
 - [16] Henikoff S. and Henikoff J.G. (1991) *Nucleic Acids Res.* 19, 6565.
 - [17] Ogiwara A., Uchiyama I., Seto Y. and Kanehisa M. (1992) *Protein Engineering* 5, 479.
 - [18] Attwood T.K. and Beck M.E. (1994) *Protein Engineering* 7, 841.
 - [19] Miller W., Boguski M.S., Raghavachari B., Zhang Z. and Hardison R.C. (1994) *J. Comput. Biol.* 1, 51.
 - [20] Lawrence C.E., Altschul S.F., Boguski M.S., Liu J.S., Neuwald A.F. and Wootton J.C. (1993) *Science* 262, 208.

- [21] Neuwald A.F. and Green P. (1994) *J. Mol. Biol.* 239, 698.
- [22] Henikoff S., (1995) in: *Biotechnology Annual Review* (El-Gewely R.M., Ed.) Vol. 1, Elsevier, Amsterdam (In press).
- [23] Bairoch A. (1992) *Nucleic Acids Res.* 20, 2013.
- [24] Pongor S., Skerl V., Cserzo M., Hatsagi Z., Simon G. and Bevilacqua V. (1993) *Protein Engineering* 6, 391.
- [25] Sonnhammer E.L.L. and Kahn D. (1994) *Prot. Sci.* 3, 482.
- [26] Wang J.T.L., Marr T.G., Shasha D., Shapiro B.A. and Chirn G.-W. (1994) *Nucleic Acids Res.* 22, 2767.
- [27] Schuler G.D., Altschul S.F. and Lipman D.J. (1991) *Proteins: Struct. Funct. Genet.* 9, 180.
- [28] Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J. (1990) *J. Mol. Biol.* 215, 403.
- [29] Schneider T.D. and Stephens R.M. (1990) *Nucleic Acids Res.* 18, 6097.
- [30] Altschul S.F., Carroll R.J. and Lipman D.J. (1989) *J. Mol. Biol.* 207, 647.
- [31] Vingron M. and Sibbald P.R. (1993) *Proc. Natl. Acad. Sci. USA* 90, 8777.
- [32] Henikoff S. and Henikoff J.G. (1994) *J. Mol. Biol.* 243, 574.
- [33] Henikoff S. (1992) *New Biol.* 4, 382.
- [34] Gribskov M., McLachlan A.D. and Eisenberg D. (1987) *Proc. Natl. Acad. Sci. USA* 84, 4355.
- [35] Bairoch A. and Boeckmann B. (1992) *Nucleic Acids Res.* 20, 2019.
- [36] Wallace J.C. and Henikoff S. (1992) *CABIOS* 8, 249.
- [37] Henikoff S. and Henikoff J.G. (1992) *Proc. Natl. Acad. Sci. USA* 89, 10915.
- [38] Pietrokovski S. (1994) *Prot. Sci.* 3, 2340.
- [39] Henikoff S. and Henikoff J.G. (1994) *Genomics* 19, 97.

- [40] Heierhorst J., Lederis K. and Richter D. (1992) *Proc. Natl. Acad. Sci. USA* 89, 6798.
- [41] Radice A.D., Bugaj B., Fitch D.H.A. and Emmons S.W. (1994) *Mol. Gen. Genet.* 244, 606.
- [42] Daniels S.B., Peterson K.R., Strausbaugh L.D., Kidwell M.G. and Chovnick A. (1990) *Genetics* 124, 339.
- [43] Doak T.G., Doerder F.P., Jahn C.L. and Herrick G. (1994) *Proc. Natl. Acad. Sci. USA* 91, 942.
- [44] Langin T., Capy P. and Daboussi M.-J. (1995) *Mol. Gen. Genet.* 246, 19.
- [45] Robertson H.M. (1993) *Nature* 362, 241.
- [46] Robertson H.M. and Lampe D.J. (1995) *Ann. Rev. Entomol.* 40, 333.
- [47] Brown M.T. (1995) *Gene* (In press).

TABLE 1.
GIBBS heuristic

Shortest sequence	N ^a	# Runs	Block Widths (W) ^b
< 36	1	12	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
36 - 100	2	6	5+11, 6+12, 7+13, 8+14, 9+15, 10+16
101 - 150	3	4	5+ 9+13, 6+10+14, 7+11+15, 8+12+16
151 - 200	4	3	5+8+11+14, 6+9+12+15, 7+10+13+16
201 - 250	5	2	5+7+9+11+13, 6+8+10+12+14
> 250	6	2	5+7+9+11+13+15, 6+8+10+12+14+16

^aNumber of Blocks

^bBlock widths for successive runs are delimited by ",", and for a single run are connected by "+".

TABLE 2.
BLIMPS/MULTIMAT search results using Tc1 blocks vs. GenBank 87
(through 2/15/95)^a

<u>Rank</u>	<u>Block</u>	<u>Acc. #</u>	<u>Species</u>	<u>Support</u> ^b	<u>Comment</u>
<u>Single- or multiple-block hits:</u>					
30	C	HIU13810	<i>H. irritans</i>		known mariner
33	D	CCGONBS1	<i>C. carpio</i>	E	new Tc1 (fish)
34	C	HIU13819	<i>H. irritans</i>		known mariner
35	C	HIU13809	<i>H. irritans</i>		known mariner
40	C	HIU13806	<i>H. irritans</i>		known mariner
41	C	HIU13818	<i>H. irritans</i>		known mariner
43	B	SMOEPDSSII	<i>S. salar</i>	A	new Tc1 (fish)
45	C	HIU13816	<i>H. irritans</i>		known mariner
48	E	SLRIBS1G	<i>X. laevis</i>	B, C	new (amphibian)
50	C	HIU13805	<i>H. irritans</i>		known mariner
51	D	ZEFEPEN	<i>B. rerio</i>	C	new Tc1 (fish)
52	C	MSQMTRNPAC	<i>A. gambiae</i>		known mariner
54	C	SCU13824	<i>S. calcitrans</i>		known mariner
55	D	RFU18764	<i>R. fredii</i>		new (plant)
57	B	S66606	<i>O. tschawytscha</i>	A	new Tc1 (fish)
58	C	HIU13803	<i>H. irritans</i>		known mariner
59	B	RATADLJE	Rat/adenovirus		first false positive
<u>Multiple-block hits only</u> ^c :					
60	E	XLXFG512 ^d	<i>X. laevis</i>	A, B	new Tc1 (amphibian)
61	A	DROINTERSP	<i>D. simulans</i>	B	new Tc1 (fly)
70	C	AGU11655	<i>A. gambiae</i>	E	known mariner
71	C	AGU11657	<i>A. gambiae</i>	D	known mariner
72	E	CCGONBS2	<i>C. carpio</i>	D	new Tc1 (fish)
99	C	HIU11641	<i>H. irritans</i>	D	known mariner
134	E	BMOLSP	<i>B. mori</i>	C	new mariner (moth)
552	B	MPOMTCG	<i>M. polymorpha</i>	E	first false positive

^aOnly hits judged to be new discoveries or known mariner sequences are reported. Four single block hits (ranking 31, 42, 46 and 53) are omitted, since they are identical to *H. irritans* hits in the aligned region.

^bFor multiple-block hits, a supporting block is one that ranked among the top 1000 in the corresponding BLIMPS search.

^cAll multiple-block hits are shown between the first false positive hit with only a single block (rank 60) and the first false positive hit involving multiple blocks (rank 552).

^dSee Fig. 3.

Fig. 1. Output from Blockmaker.

Nine full-length sequences representing members of the Tc1 family of transposons were submitted in a single message, and the results shown were returned within 2 min. For each sequence segment in a block, the position of the first amino acid is shown, with interblock distances in parentheses as indicated. Sequences are in alphabetical order. Sequences without Swiss-Prot ID designations are translations from the predicted coding regions of the following GenBank/EMBL entries: DmBari-1 for *Drosophila melanogaster* Bari-1 (X67681), DmHB1 for *D. melanogaster* HB1 (X01748), DhMinos for *D. hydei* Minos (X61695), EsTes1 for *Eptatretus stouti* Tes1 (M93038) and DhUhu for *D. heteroneura* Uhu (X17356).

Fig. 2. Example of a sequence logo.

Each stack of letters corresponds to a column in the GIBBS Tc1 Block D. Colors are: red for acidic (D, E), blue for basic (H, K, R), light grey for polar (C, S, T), green for amide (N, Q), yellow for methionine (M), black for hydrophobic (A, I, L, V), orange for aromatic (F, W, Y), purple for proline (P) and grey for glycine (G).

Fig. 3. MULTIMAT display of a hit using GIBBS Tc1 blocks in a search of GenBank 87. The query map and alignment display is modeled on the Blocksearch system described previously [39]. The hit shown lies in uncharacterized downstream sequence of XLXFG512, a GenBank entry for two *X. laevis* zinc finger genes. The P-value is calculated essentially as described [33, 39]. In the query map just below, the number of residues (92) shows the scale in amino acids, "AAA..." represents the A block roughly in proportion to its width, and colons represent the minimum and periods the maximum distance between segments in the query blocks. Colons also represent the distance between detected segments in the database sequence. Alignments are shown for each detected segment of the database sequence with the segment closest to it in the query block, where the distance between blocks (or from the beginning of the predicted amino acid sequence to the first block) is listed as (minimum, maximum): followed by the distance in the detected segment; for example, (20,24):25 means that the distance between Blocks A and B ranges from 20-24 amino acids in the 9 Tc1 sequences represented, and the corresponding distance in the database sequence is equivalent to (about) 25 amino acids. Upper case in the detected segment indicates at least one occurrence of the residue in that column of the block.

BLOCKS from MOTIF

Tc1

9 sequences are included in 4 blocks

Tc1 A, width = 46		
DmBari-1	120	KTIEITPTNKTkRLRFALEYVKKPLDFWFNlLWTDESafQYQGSYS
DmHBl	98	KVPLPSPRHIKARLSlAKTYLNWPVSKWRNlLWTDGSKIMlFGGTG
DhMinos	156	EKPLLTlRQKKKRLQWARERMSWTQRQWDTIIFsDEAKFDVSVGDT
Tc1A_CAEBR	53	KKPLVSlKNRkARVEWAKQHLsWGPREWANHlWSDESKFNMFgTDG
Tc1A_CAEL	53	KKPFISKKNRMARVAWAKAHLRWGRQEWAKHlWSDESKFNlFGSDG
Tc2A_CAEBR	53	KKPSISKKNRIARVAWARAHLHWGRQDWANHVFSDSKFNlFGTDG
Tc3A_CAEL	110	KLRPAPLLSADHKLKRlEFakNNMGtNWSKVVFSDekKFnlDGPDG
EsTesl	146	QKTIRRWQNKKKRFAWAMKHRQWTTENWKKALWTDskFEiFVSSR
DhUhu	82	KKPFISTKNKGTRMTFAKTHLDKDLFWNTIIFeDESKFIIFGSDG

Tc1 B, width = 11			Tc1 C, width = 11		
DmBari-1	(21)	187 GGGTVMFWGCL	(40)	238	WILQQDNAPCH
DmHBl	(24)	168 GGPkIMVWACF	(39)	218	WTFQQDNDQKR
DhMinos	(22)	224 FPAStMVWGCM	(40)	275	FTFQQDGASSH
Tc1A_CAEBR	(23)	122 GGGsVMVWGCF	(39)	172	WVFQQDNDPKH
Tc1A_CAEL	(23)	122 GGGsVMVWGCF	(39)	172	FVFQQDNDPKH
Tc2A_CAEBR	(23)	122 GGGsVMVWGCF	(39)	172	FVFQQDNDPKH
Tc3A_CAEL	(20)	176 GGGTVMVWGAF	(39)	226	FRFQQDNATIH
EsTesl	(22)	214 GGGSLMIWGSF	(39)	264	FILQQDNDPKH
DhUhu	(22)	150 HGGSVMVWACI	(39)	200	FRFYQDNDQKH

Tc1 D, width = 15		
DmBari-1	(18)	267 WPPQSPDLNlIENVW
DmHBl	(18)	247 WQAPPShLNPIENLY
DhMinos	(18)	304 WPSNSPDLSPiENIW
Tc1A_CAEBR	(18)	201 WPSQSPDLNPIEHMW
Tc1A_CAEL	(18)	201 WPSQSPDLNPIEHLW
Tc2A_CAEBR	(18)	201 WPSQSPDLNPIEHLW
Tc3A_CAEL	(18)	255 WPARSPDLNPIENLW
EsTesl	(22)	297 WPAQSPDLNPIELVW
DhUhu	(18)	229 XPAQSPDVNVIXNLW

Figure 1

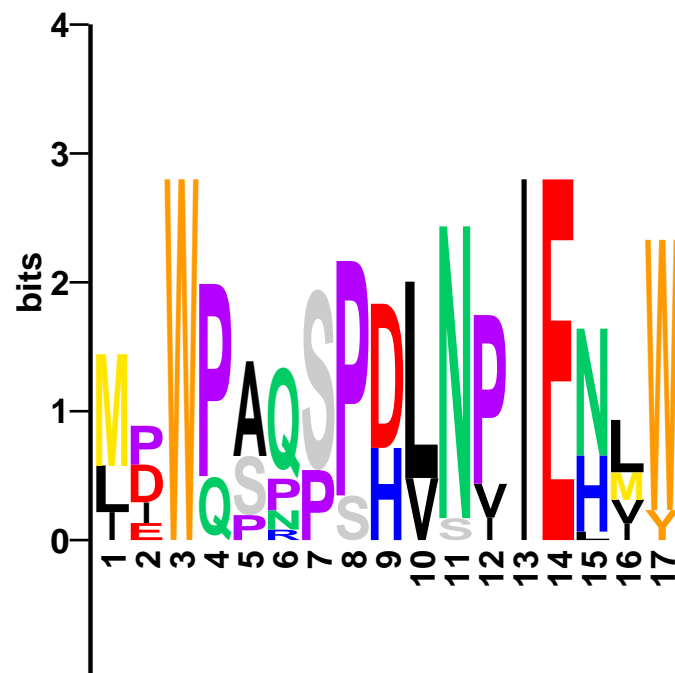
BLOCKS from GIBBS

Tc1

9 sequences are included in 5 blocks

		Tc1 A, width = 38	
DmBari-1	128	NKTKRLRFALEYVKKPLDFWFNLTDES	AFQYQGSYS
DmHb1	106	HIKARLSLAKTYLNWPVSKWRNLT	DGSKIMLFGGTG
DhMinos	164	QKKKRLQWARERMSWTQRQWD	TIIFSDEAKFDVSVGDT
TC1A_CAEBR	61	NRKARVEWAKQHLSWGP	REWANHWSDESKFNMFGTDG
TC1A_CAEL	61	NRMARVAWAKAHLRWGRQEWAK	HIWSDESKFNLFGSDG
TC2A_CAEBR	61	NRIARVAWARAHLHWGRQD	WANHVFSDESKFNLFGTDG
TC3A_CAEL	118	SADHKLKRLEFAKNNMGTN	WSKVVSDEKKFNLDGPDG
EsTes1	154	NKKKRFAWAMKHRQWTTEN	WKKALWTDESKFEIFVSSR
DhUhu	90	NKGTRMTFAKTHLKDLEFWNT	IIFEDESKFIIFGSDG
		Tc1 B, width = 11	
DmBari-1	(21)	187 GGGTVMFWGCL	(40)
DmHb1	(24)	168 GGPIMVWACF	(39)
DhMinos	(22)	224 FPASTMVWGCM	(40)
TC1A_CAEBR	(23)	122 GGGSVMVWGCF	(39)
TC1A_CAEL	(23)	122 GGGSVMVWGCF	(39)
TC2A_CAEBR	(23)	122 GGGSVMVWGCF	(39)
TC3A_CAEL	(20)	176 GGGTVMVWGAF	(39)
EsTes1	(22)	214 GGGSLMIWGSF	(39)
DhUhu	(22)	150 HGGSVMVWACI	(39)
		Tc1 C, width = 11	
DmBari-1	(21)	187 GGGTVMFWGCL	(40)
DmHb1	(24)	168 GGPIMVWACF	(39)
DhMinos	(22)	224 FPASTMVWGCM	(40)
TC1A_CAEBR	(23)	122 GGGSVMVWGCF	(39)
TC1A_CAEL	(23)	122 GGGSVMVWGCF	(39)
TC2A_CAEBR	(23)	122 GGGSVMVWGCF	(39)
TC3A_CAEL	(20)	176 GGGTVMVWGAF	(39)
EsTes1	(22)	214 GGGSLMIWGSF	(39)
DhUhu	(22)	150 HGGSVMVWACI	(39)
		Tc1 D, width = 17	
DmBari-1	(16)	265 LPWPPQSPDLNIIENVW	
DmHb1	(16)	245 MPWQAPPSHLNPIENLY	
DhMinos	(16)	302 LDWPSNSPDLSPINI	
TC1A_CAEBR	(16)	199 LEWPSQSPDLNPIEHMW	
TC1A_CAEL	(16)	199 LDWPSQSPDLNPIEHLW	
TC2A_CAEBR	(16)	199 LDWPSQSPDLNPIEHLW	
TC3A_CAEL	(16)	253 LDWPARSPDLNPIENLW	
EsTes1	(20)	295 MEWPAQSPDLNPIELVW	
DhUhu	(16)	227 IIXPAQSPDVNVIXNLW	
		Tc1 E, width = 37	
DmBari-1	(21)	303 IAEIWSKLTLEFAQTLVRS	SIPKRLQAVIDAKGGVTKY
DmHb1	(21)	283 VQDTWAKIPPKPCXDLVDF	MPRGCKAVLANKGYPKY
DhMinos	(21)	340 LQEMWDSISQEHCKNLLS	SMPKRVKCVMAKGDVTQF
TC1A_CAEBR	(21)	237 LEAAWSKIPMTVVQTLLES	MPRRCKAVIDAKGYPTKY
TC1A_CAEL	(21)	237 LENAUKAIPMSVIHKLID	SMPRRCAVIDANGYATKY
TC2A_CAEBR	(21)	237 LQDVWQAIPMSVIDTIL	SMPRRCQTVIDAKGFPTKY
TC3A_CAEL	(23)	293 ILDAWSKIPDNQLKSLVRS	MEDRLIEIIRTQGNPINY
EsTes1	(20)	332 LLQQSREELSEQYLISIV	ERMVCSAVISAKGGYFDE
DhUhu	(17)	261 LLDEWSKISPETTRKLV	SSMNNRLMEDIKAKGYHTKY

Figure 1 (continued)



Logo of Gibbs Block D (Tc1) 9 sequences

Block	Rank	Frame	Score	Location		Locus	Description
TclgibbA	27	3	2040	16056-	16167	XLXFG512	X.laavis XFG 5-1
TclgibbB	61	2	670	16244-	16274	XLXFG512	X.laavis XFG 5-1
TclgibbE	25	3	1838	16560-	16668	XLXFG512	X.laavis XFG 5-1

P<6.7e-13 for TclgibbA TclgibbB in support of TclgibbE

```

|----- 92 residues-----|
Tclgibb      AAAAAAAAAA:.....BBB:.....CCC:.....DDDD:.....EEEEEEEEEE
XLXFG512     AAAAAAAAAA:.....BBB:.....EEEEEEEEEE

```

```

TclgibbA      <->A      (60,163):5351
DmBari-1      128      NKTKRLRFALEYVKKPLDFWFWNILWTDESAFYQGSYS
|             | | |      | | | | | | | | | |
XLXFG512      5352      NtKAtLdFAKKtsKKaaQlWKNIVWTDetKmkLQqnDG

```

```

TclgibbB      A<->B      (20,24):25
TC1A_CAEBR    122      GGSVMVWGCF
| | | | | | | |
XLXFG512      5415      lGSVMaWGAw

```

```

TclgibbE      B<->E      (100,111):94
DhMinos       340      LQEMWDSISQEHCKNLLSSMPKRVKCVMAKGDVTQF
| | | | | | | | | | | | | |
XLXFG512      5520      avkTWQSIkQEQTHNLLmSMsstLQpVIAAKaFtpti

```

Figure 3